Efficient Detection of a Continuous Wave Signal with a Linear Frequency Drift

Paul N. Swarztrauber and David H. Bailey

RNR Technical Report RNR-90-002

Efficient detection of a continuous-wave signal with a linear frequency drift.

Paul N. Swarztrauber [1,3] and David H. Bailey [2]

January 24, 1990

ABSTRACT

An efficient method is presented for the detection of a continuous-wave (CW) signal with a frequency drift that is linear in time. Signals of this type occur if the transmitter and receiver are accelerating relative to one another, e.g. transmissions from the Voyager space craft. We assume that both the frequency and the drift are unknown. We also assume that the signal is weak compared to the Gaussian noise. The signal is partitioned into subsequences whose discrete Fourier transforms provide a sequence of instantaneous spectra at equal time intervals. These spectra are then accumulated with a shift that is proportional to time. When the shift is equal to the frequency drift, the signal to noise ratio increases and detection occurs. In this paper we show how to compute these accumulations for many shifts in an efficient manner using a variant of the FFT. Computing time is proportional to $L\log L$ where L is the length of the time series. Computational results are presented.

1 National Center for Atmospheric Research, Boulder, Colorado 80307, which is sponsored by the National Science Foundation.

2 NASA Ames Research Center, Moffett Field, California 94035.

3 This work was supported by the NAS Systems Division via Cooperative Agreement NCC 2-387 between NASA and the University Space Research Association (USRA). This work was performed while this author was visiting the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffett Field, CA 94035.

1. Introduction

For $l=0,...,L-1$ we are given a time series y_l that is is generated from a weak continuous wave (CW) signal whose frequency is drifting linearly with time, i.e.,

$$y_l = Ae^{2\pi i(\omega_0 + \omega_1 t_l)t_l} + \sigma g_l. \tag{1.1}$$

The frequency at time $t_0 = 0$ is ω_0 and the frequency drift is ω_1. Both are measured in cycles per unit time. The term σg_l is Gaussian noise with mean zero and standard deviation σ.

Signals of the form (1.1) result from CW transmissions between any locations that are accelerating with respect to one another, e.g. like those received from the Voyager space craft. They are also of interest for applications where the acceleration is considerably reduced but a high level of accuracy is required, such as satellite based aircraft navigational systems and similar systems that are currently under development for automobiles. Signals of this form are also thought to be the most probable form of initial communication from extraterrestrials [3].

THE PROBLEM:

> Given the time series y_l with a low signal to noise ratio then determine ω_0 and ω_1.

Once a signal is detected, its verification and analysis is relatively simple using tuned versions of the algorithm that will be presented in the later sections. Therefore detection becomes the fundamental problem and the primary goal is to maximize its likehood. With a fixed computing resource the following options are available:

1. Develop a new method of detection.
2. Speed the computation of an existing method and use higher resolution.
3. Improve the accuracy of the existing method.

In this paper we will pursue options 2 and 3 in the context of an existing method which is described in [3] and [4]. They use the fast Fourier transform (FFT) to compute, but not to accumulate the spectra. Here we show that the FFT can be used in all of phases which results in a significant reduction in computing time. In particular the FFT is used to compute certain spectral shifts with a high degree of accuracy at the same time that it is used to accumulate the shifts.

The approach in Cullers, et.al., [3] is to accumulate the spectra of subsequences to increase the signal to noise ratio. If L has factors $L = MN$ we can define M subsequences $y_{m,n} = y_l$ where $l = n + mN$, for $m = 0,...,M-1$ and $n = 0,...,N-1$. Each subsequence has N elements. Without loss of generality we can assume that the sequence y_l is tabulated at integer times $t_l = l = n + mN$ and consequently (1.1) takes the form

$$y_{m,n} = Ae^{2\pi i(\omega_0 + mN\omega_1)mN} e^{2\pi i(\omega_0 + 2mN\omega_1 + n\omega_1)n} + \sigma g_{m,n}. \qquad (1.2)$$

At this point, the sum of these subsequences would not increase the signal to noise ratio because $y_{m,n}$ is not coherent as a function of m. Nevertheless, the following steps can be taken:

1. Compute the discrete Fourier transform (DFT) of the subsequences, i.e., compute M row transforms of length N

$$Y_{m,k} = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} y_{m,n} e^{-ikn\frac{2\pi}{N}} \qquad (1.3)$$

If the frequency drift is negligible on each subsequence, i.e. if $n\omega_1 \ll \omega_0 + 2mN\omega_1$, then $Y_{m,k}$ provides an approximation to the "instantaneous" spectrum at time mN. If we set $n\omega_1 = 0$ in (1.2) and $k = N(\omega_0 + 2mN\omega_1)$ in (1.3) then

$$Y_{m,N(\omega_0 + 2mN\omega_1)} = Ae^{2\pi i(\omega_0 + mN\omega_1)mN} + noise \qquad (1.4)$$

The contribution of the signal to $Y_{m,k}$ is maximum at $k = N(\omega_0 + 2mN\omega_1)$ because the terms in (1.3) are coherent.

2. Next compute the squared amplitude (power) of the spectrum $X_{j,k} = Y_{j,k}\overline{Y}_{j,k}$. Then

$$X_{m,N(\omega_0 + 2mN\omega_1)} = |A|^2 + noise. \tag{1.5}$$

The purpose of this step is to provide an "instantaneous" spectra that is coherent for the next step.

3. Finally the spectra $X_{j,k}$ is summed with a "test" shift α ,i.e.

$$S_k(\alpha) = \sum_{m=0}^{M-1} X_{m,k+\alpha m}. \tag{1.6}$$

From (1.5) and (1.6) the signal to noise ratio is maximum for $k + \alpha m = N(\omega_0 + 2mN\omega_1)$ or for $k = N\omega_0$ and $\alpha = 2N^2\omega_1$. Therefore the problem becomes one of computing the k and α that correspond to the maximum of $S_k(\alpha)$ since ω_0 is then given by $\omega_0 = k/N$ and ω_1 is given by $\omega_1 = \alpha/(2N^2)$. To implement (1.6) it is necessary to interpolate the spectra to non-integer values $k+\alpha m$ of the subscript. An accurate method for this interpolation is given in the next section.

The maximum of (1.6) is located by tabulating $S_k(\alpha)$ at a set of points $\alpha_j = \alpha + j\delta$ for $j=0,...,M$. Accuracy can be improved by a second tabulation in the neighborhood of the maximum of the first tabulation which can be used to verify detection. If we set $S_{j,k} = S_k(\alpha_j)$ then in the sections that follow we will be concerned with the efficient tabulation of a discrete version of (1.6) namely (2.1).

2. The Algorithm.

Given α, δ and the $M \times N$ matrix $X_{m,n}$, then we wish to compute another $M \times N$ matrix

$$S_{j,k} = \sum_{m=0}^{M-1} X_{m,k+m(\alpha+j\delta)}. \tag{2.1}$$

The jth row of $S_{j,k}$ contains the sum of the rows of $X_{m,k}$ where the mth row is shifted by $m(\alpha+j\delta)$. The smallest shift between any two adjacent rows is δ. Fractional shifts are permitted and computed with a high degree of accuracy using trigonometric interpolation and the FFT.

Let $x_{m,n}$ be the rowwise forward Fourier transform of $X_{m,k}$. Then

$$X_{m,k} = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} x_{m,n} e^{ikn\frac{2\pi}{N}} \tag{2.2}$$

and

$$X_{m,k+m(\alpha+j\delta)} = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} x_{m,n} e^{i[k+m(\alpha+j\delta)]n\frac{2\pi}{N}}. \tag{2.3}$$

From (2.1)

$$S_{j,k} = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} \left[\sum_{m=0}^{M-1} (e^{imn\alpha\frac{2\pi}{N}} x_{m,n}) e^{i\delta jmn\frac{2\pi}{N}} \right] e^{ikn\frac{2\pi}{N}} \tag{2.4}$$

or

$$S_{j,k} = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} s_{j,n} e^{ikn\frac{2\pi}{N}} \tag{2.5}$$

where

$$s_{j,n} = \sum_{m=0}^{M-1} z_{m,n} e^{i\delta jmn\frac{2\pi}{N}} \tag{2.6}$$

and

$$z_{m,n} = e^{imn\alpha\frac{2\pi}{N}} x_{m,n}. \tag{2.7}$$

Given α, δ, and $X_{m,k}$, the algorithm consists of the following steps:

A. Compute $x_{m,n}$ from the inverse of (2.2) and $z_{m,n}$ from (2.7).

B. Next compute $s_{j,n}$ using (2.6). With fixed n, each column can be transformed independently using a method that will be presented below.

C. Compute $S_{j,k}$ from (2.5) using the backward FFT applied to each row.

D. If $S_{jmax,kmax} = \max_{j,k} S_{j,k}$ then $\omega_0 = kmax/N$ and $\omega_1 = (\alpha + jmax\delta)/(2N^2)$

The FFT can be used for steps A and C. However it cannot be used for step B since δ is not necessarily an integer. Nevertheless we will show that (2.6) can be computed in about the time required by a fast M-point linear convolution. We will use a technique introduced by Bluestein [2] that enabled him to develop an FFT for arbitrary M including large prime numbers. The details of this algorithm are presented in [5] as well as its implementation on multiprocessors. Bluestein's technique has also been used to develop a fast "fractional" Fourier transform which is described in detail in [1].

The same transform (2.6) is applied to each column of the array $z_{m,n}$ and hence the exposition can be simplified by removing the subscript n. Therefore we wish to compute

$$s_j = \sum_{m=0}^{M-1} z_m e^{i\beta jm\frac{2\pi}{N}} \tag{2.8}$$

where $\beta = \delta n$. We begin with the following identity

$$2jm = j^2 + m^2 - (j-m)^2 \tag{2.9}$$

Substituting into (2.8) we obtain

$$s_j = e^{i\beta j^2 \frac{\pi}{N}} \sum_{m=0}^{M-1} (e^{i\beta m^2 \frac{\pi}{N}} z_m) e^{-i\beta (j-m)^2 \frac{\pi}{N}} \tag{2.10}$$

or

$$s_j = b_j \sum_{m=0}^{M-1} b_{|j-m|}^{-1} c_m \tag{2.11}$$

where

$$b_m = e^{i\beta m^2 \frac{\pi}{N}} \tag{2.12}$$

$$c_m = b_m z_m \tag{2.13}$$

Equation (2.11) is preferable to (2.8) for computing s_j because (2.11) can be evaluated efficiently using the FFT. However an indirect approach is required because (2.11) is not a circular convolution, i.e., the subscript $j-m$ is not interpreted modulo M. If $j-m$ is negative then it is replaced by $m-j$ rather than $j-m+M$. Nevertheless, one can construct an equivalent circular convolution.

Select $P \geq 2M-2$ as a highly factorizable integer, e.g., a power of two. Next, extend b_m and c_m to sequences with P elements as follows:

$$b_m = 0 \qquad M \leq m < P-M \tag{2.15a}$$

$$b_m = b_{P-m} \qquad P-M \leq m < P. \tag{2.15b}$$

$$c_m = 0 \qquad M \leq m < P \tag{2.15c}$$

Define

$$\hat{s}_j = b_j^{-1} \sum_{m=0}^{P-1} b_{j-m} c_m \tag{2.16}$$

where the subscripts are now interpreted modulo P. It can be determined by inspection that $s_j = \hat{s}_j$ for $j = 0,...,M-1$. Since (2.16) is a P point circular convolution, it can be evaluated using the familiar FFT procedure [6]. The algorithm for fast evaluation of (2.11) can now be summarized.

B.0 Compute b_m from (2.12) and extend to length P using (2.15a) and (2.15b). Perform a P point forward FFT and call the result B_k.

B.1 Compute c_m from (2.13) and extend to length P using (2.15c). Perform a P point forward FFT and call the result C_k.

B.2 Compute $D_k = B_k C_k$. Perform a backward P-point FFT and call the result d_k.

B.3 Multiply the first M elements of d_k by b_j^{-1} to obtain $s_j = \hat{s}_j$. The remaining $P-M$ elements of d_k may be discarded.

Note that step B.0 is an initialization step that does not have to be repeated for the analysis of any subsequent time series.

3. Computational Results

The algorithm described above has been implemented for test purposes. The resulting program is written entirely in Fortran without assembly code or assembly-coded library routines. Some modest effort was made to insure that the code was suitable for reasonably high performance execution on vector computers, but in general it is an entirely straightforward implementation of the algorithm presented in section 2.

Two details of this implementation are worth mentioning because they would be worth inclusion in any serious production system. First of all, every FFT operation, including those that are a part of the fractional FFT computation, was performed using "simultaneous" FFT routines. In other words, the FFTs were performed vector-wise instead of individually. Such an implementation is of course highly suited for a vector computer, but it also is quite appropriate for implementation on a parallel computer system. Secondly, it is important to note that a substantial amount of computation time can saved by employing real-to-complex FFTs in step A above and by employing complex-to-real FFTs in step C above. A by-product of this modification is that the FFTs required for the fractional transforms in step B need only be performed on $N/2+1$ columns.

The detection of the true signal base frequency and drift rate is enhanced in this implementation by employing a weighted score. This is because even when a pure signal with no noise is input to the processing algorithm, the peak in the result array is "smeared" over a number of nearby elements. This pattern of smearing from a pure signal can in fact be used to design an improved detection filter. The authors found that the following weighted score formula proved both inexpensive and effective:

$$T_{j,k} = 1.052S_{j-2,k+1} + S_{j-1,k} + S_{j-1,k+1} + 1.216S_{j,k} + S_{j+1,k-1} + S_{j+1,k} + 1.052S_{j+2,k-1} \quad (3.1)$$

where $S_{j,k}$ is the final result of step C above. The statistical significances of these weighted scores are determined by explicitly computing the mean and standard deviation of all $T_{j,k}$.

Table 1 contains a detailed accounting of the computational cost of the complete detection algorithm. "C-C", "R-C" and "C-R" denote the three types of FFTs: complex-to-complex, real-to-complex, and complex-to-real. The column headed "Ref." lists references to specific equations and algorithm steps. The column headed "Operation Counts" contains the number of real floating point arithmetic operations for each step, with adds, subtracts and multiplies each counting as one operation.

Table 1
Floating point operation counts

Computational Step	Ref.	Operation Count
Initial row-wise C-C FFTs	(1.3)	$5MN\log_2 N$
Amplitude squared (power)	(1.5)	$3MN$
Forward row-wise R-C FFTs	A	$(5MN\log_2 N)/2+MN$
Column-wise multiplications	B.1	$3MN$
Column-wise forward C-C FFTs	B.1	$5MN\log_2 M+5MN$
Pointwise multiplications	B.2	$6MN$
Column-wise inverse C-C FFTs	B.2	$5MN\log_2 M+5MN$
Column-wise multiplications	B.3	$3MN$
Backward row-wise C-R FFTs	C	$(5MN\log_2 N)/2+MN$
Weighted scores and statistics	(3.1)	$13MN$
Total		$MN(10\log_2 MN+55)$

This algorithm was exercised by performing a series of tests with pseudorandomly selected initial frequencies and drift rates. In these tests, M and N were each set to 1,024. The initial value of the drift rate α was set to $-1/2$, and the drift rate increment, δ, was set to $-1/M$. Thus the sums $S_k(\alpha)$ in (1.6) were computed for all drift rates α from $-1/2$ to 1/2. For each test, a linearly drifting signal was added to complex Gaussian pseudorandom data of the form $g_l = (e_l+f_l)/\sqrt{2}$, where e_l and f_l are real Gaussian data with zero mean and unit variance. The ratio of the amplitudes of the noise and the signal was 56.6, so that the power of the signal was lower than that of the noise by a ratio of 3,200. This extremely low signal to noise level is approximately the level that the Cyclops SETI system hopes to detect [3].

Table 2
Algorithm Test Results

Trial No.	Generated		Detected		Z-score	Time
	Base Freq.	Drift Rate	Base Freq.	Drift Rate		
1	119.842	-0.26187	119.00	-0.2607	7.812	2.342
2	967.298	0.49238	968.00	0.4922	7.266	2.423
3	706.734	0.04648	706.00	0.0479	6.976	2.364
4	828.500	0.05203	828.00	0.0537	7.177	2.293
5	97.555	0.35731	98.00	0.3564	7.124	2.430
6	626.842	-0.25255	627.00	-0.2529	6.485	2.472
7	989.664	-0.14073	989.00	-0.1396	7.836	2.476
8	203.058	-0.42424	203.00	-0.4248	8.902	2.592
9	1022.170	-0.10787	1022.00	-0.1074	8.451	2.315
10	921.632	0.43326	922.00	0.4336	10.460	2.550
Ave.					7.849	2.429

These tests were performed on the Cray-2 operated by the NAS Systems Division at NASA Ames. The results of these tests are displayed in Table 2. The column headed Base Freq. is $N\omega_0$ and the column headed Drift Rate is $2N^2\omega_1$. The column headed "Z-score" gives the Z-score of the detection, i.e. the number of standard deviations above the mean. The column headed "Time" contains single processor CPU times in seconds. These results indicate that the detection algorithm is effective in determining the unknown base frequency and drift rates. In every trial, the element of the final array with the highest weighted score was within one bin both in drift rate and frequency from the true value. These Z-scores are well above the level of random scores since one would not expect a random Z-score to exceed 5 in a 1,024×1,024 array.

The average processing CPU time, 2.429 seconds, corresponds to a performance rate of approximately 109 MFLOPS, based on the total number of floating point operations given in Table 1. This performance rate could be increased by employing assembly-coded simultaneous FFT routines and by employing all four

of the Cray-2 processors. In any event, these timings indicate that such computations are feasible given suitably fast computer hardware.

As was mentioned above, the central problem is making an accurate initial detection. The algorithm presented here accomplishes this basic objective. Once a putative detection has been made, its confidence level can be increased in a number of ways, such as by repeating our algorithm with higher resolution near the putative detection, or by employing a matched filter keyed to the putative frequency and drift rate. It is also expected that some modest enhancements and improvements in the algorithm could also improve these detection levels.

4. Acknowledgements We wish to thank Dr. Palmer Dyal for bringing the SETI problem to our attention and Drs. Kent Cullers and Peter Backus for providing the details of the problem and reference [3].

References

1. D.H. Bailey and P.N. Swarztrauber, The fast fractional Fourier transform and its applications, *SIAM Rev.*, submitted for publication.
2. L.I. Bluestein, A linear filtering approach to the computation of the discrete Fourier transform, *IEEE Trans. Audio Electroacoust.*, vol AU-18, 4(1970), pp. 451-455. Reprinted in : Digital Signal Processing, ed. L.R. Rabiner and C.M. Rader, pp. 317-321, IEEE press, New York, 1972.
3. D.K. Cullers, I.R. Linscott, and B.M. Oliver, Signal processing in SETI,*Comm. ACM*, 28(1985), pp. 1151-1163.
4. I. Gertner, Optimal detection and separation of chirp signals, *IEEE Int. Conf. Acoust. Speech Signal Proc.*, April 3-4, 1990, Albuquerque, NM.
5. H.J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*, Springer-Verlag, New York, 1981.
6. P.N. Swarztrauber, R.A. Sweet, W.L. Briggs, V.E. Henson and J. Otto, Bluestein's FFT for arbitrary N on the hypercube, *Parallel Computing*, submitted for publication.